UNITED STATES PATENT AND TRADEMARK OFFICE

m.N

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,054	02/26/2004	Yukio Oguma	122.1582	3298

21171 7590 06/20/2007

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
RAHMAN, FAHMIDA	
ART UNIT	**PAPER NUMBER**
2116	
MAIL DATE	**DELIVERY MODE**
06/20/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/786,054	OGUMA, YUKIO	
	Examiner	Art Unit	
	Fahmida Rahman	2116	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 May 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to communications filed on 5/10/2007.

2. Claims 1, 5, 12, 17, 18 have been amended, claim 2 has been canceled and no new claims have been added. Thus, claims 1, 3-18 are pending.

Claim Objections

Claim 17 is objected to because of the following informalities: "said access boot device" in line 17 should be changed to –said accessed boot device--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites the limitation "said variable data" in line 18. There are three variable data recited in lines 5-8 of claim 1. It is unclear which variable data is referred in line 18 of claim 1. It is necessary to establish a relationship among the recited variable data. For the rest of the action, it is assumed that –said second variable data— was intended.

Claims 3-16 depend on claim 1. Thus, they carry the same ambiguity of claim 1.

Claim 1 further recites --said booting order included in said device setting data—in line 15. According to lines 1-9, booting order is defined by second variable data and device setting data is defined in first variable data. It is unclear how booting order is included in said device setting data. For the rest of the action, it is assumed that "a boot device of a first boot candidate included in said device setting data is accessed according to said booting order" was intended.

Claims 17 and 18 recite --said booting order included in said device setting data—in line 13. According to lines 1-9, booting order is defined by second variable data and device setting data is defined in first variable data. It is unclear how booting order is included in said device setting data. For the rest of the action, it is assumed that "a boot device of a first boot candidate included in said device setting data is accessed according to said booting order" was intended.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee et al (US Patent 6754818), in view of Applicant's Admission of Prior Art (AAPA).

For claim 1, Lee et al teach the following limitations:

An apparatus (Fig 1) **where an operating system** (lines 13-17 of column 3) **read out from a selected device of a multiplexed plurality of devices** (102) **is started up** (260) **for starting up the system** (abstract), **comprising: a storing unit** (110) **which stores environment data** (118, 114, 120) **for setting a boot from said plurality of devices** (Fig 2), **said environment data includes first variable data** (114 in Fig 1) **including device setting data designating a boot candidate for said plurality of devices** (lines 28-31 of column 3), **second variable data** (120 in Fig 1) **including index data setting a booting order of boot candidates** (lines 1-5 of column 4 mention that a different boot device is selected based on SBDID and list of available boot devices. Thus, previous SBDID is used to update new SBDID. Therefore, SBDID sets a booting order of the boot candidates) **set by said device setting data** (new SBDID is selected based on previous SBDID and list of available boot devices; lines 50-55 of column 3) **and third variable data** (118 in Fig 1) **indicating whether said multiplexing is valid or not is set** (lines 45-55 of column 3 mentions that BISST performs a number of operations including a list of boot devices available. Lines 15-20 of column 4 mention that successful booting is performed as long as at least one uncorrupted boot image available. Therefore, BISST notifies the system when

uncorrupted image is available by maintaining the available boot device list, which indicates multiplexing is valid. When no more uncorrupted image is available, the multiplexing is invalid); **a boot control unit** (106 and 108) **which decides on a boot device** (250 in Fig 2) **based on the setting of first variable data, second variable data and third variable data included in said environment data** (Fig 2 shows that 250 depends on 210-240 that makes use of 114, 118, 120) **and starting up said operating system** (260), **and a control unit** (Fig 2 is a control routine. Thus, there is an associated control unit to execute the routine) **which controls multiplexing of said plurality of devices, said control unit changing the setting of said index data included in said second variable data** (240) **when an abnormality is detected in said boot device** (lines 48-50 of column 2 mention that another boot image is selected when computer system hangs on a corrupted image) **and clearing said variable data to an initial value when booting is successful** (lines 5-10 of column 4 mention that round robin fashion is used for boot device selection. Therefore, successful boot also clears the variable data for subsequent boot attempt; lines 20-25 of column 6), **and said boot control unit switching said accessed boot device to another boot device according to said changed index data and controlling a boot of the other boot device** (lines 1-20 of column 4 mention that different boot device is selected based on updated SBDID if the system hangs during bootstrapping. System is booted if the different boot device has uncorrupted boot device available).

Lee et al teaches the third variable data (118) where whether multiplexing is valid or not is set. Lee et al do not explicitly mention that said control unit setting “valid” in said third variable data when a boot device of a first boot candidate is accessed according to said booting order included in said device setting data.

However, these limitations are within the spirit of Lee et al. The control unit accessed a boot device according to 114 and 120. If the boot device of first boot candidate in 114 is accessed according to said booting order in 120, and the corresponding booting is successful, the boot device has uncorrupted image. In this case, multiplexing is valid. As corrupted images are removed from device list (lines 15-20 of column 6), part of BISST 118, or the third variable data, that maintains list of available devices (lines 45-50 of column 3), can be thought as set to “valid” as available device list is non-empty due to successful booting of first candidate. Successful booting of first boot device indicates available device list will be continued as non-empty (since no removal of corrupted device takes place) for next booting, which indicates the multiplexing is valid. Therefore, BISST 118 that is responsible for maintaining available device list is set to valid when there is non-empty list and the non-empty list is ensured by successful booting of first boot candidate.

Lee et al starts up the OS from one of the boot devices (lines 10-20 of column 3 mention that OS stored in boot devices and lines 10-15 of column 5 mention that OS loaded based on instructions in the boot image. Lines 10-14 of column 3 mention that

boot devices 102 contain one or more copies of operating system. 260 of Fig 2 shows the loading of OS. Therefore, it is one of the boot devices 102 that stores the OS that is loaded in step 260. Since the system of Lee permits rotation of boot devices, the device containing OS can be selected as the boot device (as shown in 250 of Fig 2) to load the boot image from and read out the OS from that device), although Lee et al do not explicitly mention about starting of OS stored in the boot device decided by the boot control unit.

AAPA teaches an apparatus (Fig 7) where an operating system read out from a selected device of a multiplexed plurality of devices (D1, D2) is started up for starting up the system, comprising: a boot control unit (M1, M2) which decides on a boot device ([0044]) based on the setting of variable data included in said environment data (NM2) and starting up said operating system ([0047]).

It would have been obvious for one ordinary skill in the art at the time the invention was made to combine the teachings of Lee et al and AAPA. One ordinary skill in the art would be motivated to start OS stored in the boot device decided by the boot control unit, since storing OS in boot device and reading from them makes the control easier than reading OS from another location. Therefore, ordinary skill would select the boot device not only for loading boot image, but also for loading the OS from the selected device.

For claim 3, the system of Lee et al follows the round robin approach. Thus, the device initially set in device setting data can be selected for booting. In addition, system clears the index data shown in 240.

For claim 7, system of Lee et al boots up when BISST returns a good selected boot device (i.e., multiplexing is valid) and index data changes to reflect the current boot device. The initial value of index data is the previous boot device.

For claim 8, system of Lee et al reports corrupted image and ensures booted into a good state (lines 14-15 of column 6).

For claim 9, second variable data is cleared to initial value at the beginning.

For claim 10, BISST returns “no” or false when there is no device available (i.e., plurality of devices are not set for redundant operation).

For claim 11, lines 16-19 of column 6 mention that corrupted boot image is removed until it is repaired. Fig 5 shows that only the current boot device is connected to BDP. Thus, the earlier failed boot device is cut off and a new device is connected to boot port.

For claim 12, 320 is a non-volatile memory and the settings can be rewritten.

For claim 13, 112 is the boot firmware stored in 110.

For claim 14, control unit of Lee et al executes controlling of multiplexing of the plurality of devices and switches to another device when an abnormality has occured. However, Lee et al does not require system software read out from boot device to check multiplexing and processing. Applicant admits that the system software read out from boot disk controls multiplexing and switching the booted disk drive ([0039] and [0040] of page 12 of applicants disclosure). One ordinary skill in the art would be motivated to control multiplexing based on system software read out from boot device, since that would provide the redundancy of system software. As system software is saved in all devices, failure of BISST does not prevent system from working.

For claim 15, 260 shows the loading and initialization of operating system.

For claim 17, Lee et al teach the following limitations:

A method for starting up data processing system in which (Fig 1) **an operating system read out from a selected device of a multiplexed plurality of devices** (104) **is started up for starting up the system** (abstract), **comprising: storing** (110) **environment data** (118, 114, 120) **for setting a boot from said plurality of devices** (Fig 2), **said environment data includes first variable data** (114 in Fig 1) **including device setting data designating a boot candidate for said plurality of devices** (lines 28-31 of column 3), **second variable data** (120 in Fig 1) **including index data**

setting on a booting order of boot candidates set by said device setting data (lines 50-55 of column 3) **and third variable data** (118 in Fig 1) **indicating whether said multiplexing is valid or not is set** (lines 45-55 of column 3 mentions that BISST performs a number of operations including a list of boot devices available. Lines 15-20 of column 4 mention that successful booting is performed as long as at least one uncorrupted boot image available. Therefore, BISST notifies the system when uncorrupted image is available by maintaining the available boot device list, which indicates multiplexing is valid. When no more uncorrupted image is available, the multiplexing is invalid); **deciding on a boot device** (240) **based on the setting of said environment data** (210, 220) **and starting up said operating system stored in said boot device** (260), **and controlling** (Fig 2 is a control routine. Thus, there is an associated control unit to execute the routine) **multiplexing of said plurality of devices and changing the setting of said index data included in said second variable data** (240) **when an abnormality is detected in said accessed boot device** (lines 48-50 of column 2 mention that another boot image is selected when computer system hangs on a corrupted image) **and clearing said variable data to an initial value when booting is successful** (lines 5-10 of column 4 mention that round robin fashion is used for boot device selection. Therefore, successful boot also clears the variable data for subsequent boot attempt; lines 20-25 of column 6), **and switching said accessed boot device to another boot device according to said changed index data and controlling a boot of the other boot device** (lines 1-20 of column 4 mention that different boot device is selected based on updated SBDID if the system

hangs during bootstrapping. System is booted if the different boot device has uncorrupted boot device available).

Lee et al teaches the third variable data (118) where whether multiplexing is valid or not is set. Lee et al do not explicitly mention *setting "valid" in said third variable data when a boot device of a first boot candidate is accessed according to said booting order included in said device setting data.*

However, these limitations are within the spirit of Lee et al. The control unit accessed a boot device according to 114 and 120. If the boot device of first boot candidate in 114 is accessed according to said booting order in 120, and the corresponding booting is successful, the boot device has uncorrupted image. In this case, multiplexing is valid. As corrupted images are removed from device list (lines 15-20 of column 6), part of BISST 118, or the third variable data, that maintains list of available devices (lines 45-50 of column 3), can be thought as set to "valid" as available device list is non-empty due to successful booting of first candidate. Successful booting of first boot device indicates available device list will be continued as non-empty (since no removal of corrupted device takes place) for next booting, which indicates the multiplexing is valid. Therefore, BISST 118 that is responsible for maintaining available device list is set to valid when there is non-empty list and the non-empty list is ensured by successful booting of first boot candidate.

Lee et al starts up the OS from one of the boot devices (lines 10-20 of column 3 mention that OS stored in boot devices and lines 10-15 of column 5 mention that OS loaded based on instructions in the boot image. Lines 10-14 of column 3 mention that boot devices 102 contain one or more copies of operating system. 260 of Fig 2 shows the loading of OS. Therefore, it is one of the boot devices 102 that stores the OS that is loaded in step 260. Since the system of Lee permits rotation of boot devices, the device containing OS can be selected as the boot device (as shown in 250 of Fig 2) to load the boot image from and read out the OS from that device), although Lee et al do not explicitly mention about starting of OS stored in the boot device decided by the boot control unit.

AAPA teaches a method (Fig 7) for starting up data processing system in which an operating system read out from a selected device of a multiplexed plurality of devices (D1, D2) is started up for starting up the system, comprising: deciding on a boot device ([0044]) based on the setting of variable data included in said environment data (NM2) and starting up said operating system ([0047]).

It would have been obvious for one ordinary skill in the art at the time the invention was made to combine the teachings of Lee et al and AAPA. One ordinary skill in the art would be motivated to start OS stored in the boot device decided by the boot control unit, since storing OS in boot device and reading from them makes the control easier than reading OS from another location. Therefore, ordinary skill would select the boot

device not only for loading boot image, but also for loading the OS from the selected device.

For claim 18, Lee et al teach the following limitations:

A recording medium storing a program for starting up data processing system (Fig 1) **in which an operating system read out from a selected device of a multiplexed plurality of devices** (104) **is started up for starting up the system (abstract), comprising the steps of: storing** (110) **environment data** (118, 114, 120) for setting a boot from said plurality of devices (Fig 2), **said environment data includes first variable data** (114 in Fig 1) **including device setting data designating a boot candidate for said plurality of devices** (lines 28-31 of column 3), **second variable data** (120 in Fig 1) **including index data designating a boot device based on said device setting data** (lines 50-55 of column 3) **and third variable data** (118 in Fig 1) **indicating whether said multiplexing is valid or not is set** (lines 45-55 of column 3 mentions that BISST performs a number of operations including a list of boot devices available. Lines 15-20 of column 4 mention that successful booting is performed as long as at least one uncorrupted boot image available. Therefore, BISST notifies the system when uncorrupted image is available by maintaining the available boot device list, which indicates multiplexing is valid. When no more uncorrupted image is available, the multiplexing is invalid); **deciding on a boot device** (240) **based on the setting of said environment data** (210, 220) **and starting up said operating system stored in said boot device** (260), **and controlling** (Fig 2 is a control routine. Thus, there is an

associated control unit to execute the routine) **multiplexing of said plurality of devices and changing the setting of said index data included in said second variable data** (240) **when an abnormality is detected in said accessed boot device** (lines 48-50 of column 2 mention that another boot image is selected when computer system hangs on a corrupted image) **and clearing said variable data to an initial value when booting is successful** (lines 5-10 of column 4 mention that round robin fashion is used for boot device selection. Therefore, successful boot also clears the variable data for subsequent boot attempt; lines 20-25 of column 6), **and switching said accessed boot device to another boot device according to said changed index data and controlling a boot of the other boot device** (lines 1-20 of column 4 mention that different boot device is selected based on updated SBDID if the system hangs during bootstrapping. System is booted if the different boot device has uncorrupted boot device available).

Lee et al teaches the third variable data (118) where whether multiplexing is valid or not is set. Lee et al do not explicitly mention *setting "valid" in said third variable data when a boot device of a first boot candidate is accessed according to said booting order included in said device setting data.*

However, these limitations are within the spirit of Lee et al. The control unit accessed a boot device according to 114 and 120. If the boot device of first boot candidate in 114 is accessed according to said booting order in 120, and the corresponding booting is successful, the boot device has uncorrupted image. In this case, multiplexing is valid.

As corrupted images are removed from device list (lines 15-20 of column 6), part of BISST 118, or the third variable data, that maintains list of available devices (lines 45-50 of column 3), can be thought as set to "valid" as available device list is non-empty due to successful booting of first candidate. Successful booting of first boot device indicates available device list will be continued as non-empty (since no removal of corrupted device takes place) for next booting, which indicates the multiplexing is valid. Therefore, BISST 118 that is responsible for maintaining available device list is set to valid when there is non-empty list and the non-empty list is ensured by successful booting of first boot candidate.

Lee et al starts up the OS from one of the boot devices (lines 10-20 of column 3 mention that OS stored in boot devices and lines 10-15 of column 5 mention that OS loaded based on instructions in the boot image. Lines 10-14 of column 3 mention that boot devices 102 contain one or more copies of operating system. 260 of Fig 2 shows the loading of OS. Therefore, it is one of the boot devices 102 that stores the OS that is loaded in step 260. Since the system of Lee permits rotation of boot devices, the device containing OS can be selected as the boot device (as shown in 250 of Fig 2) to load the boot image from and read out the OS from that device), although Lee et al do not explicitly mention about starting of OS stored in the boot device decided by the boot control unit.

AAPA teaches a method (Fig 7) for starting up data processing system in which an operating system read out from a selected device of a multiplexed plurality of devices (D1, D2) is started up for starting up the system, comprising: deciding on a boot device ([0044]) based on the setting of variable data included in said environment data (NM2) and starting up said operating system ([0047]).

It would have been obvious for one ordinary skill in the art at the time the invention was made to combine the teachings of Lee et al and AAPA. One ordinary skill in the art would be motivated to start OS stored in the boot device decided by the boot control unit, since storing OS in boot device and reading from them makes the control easier than reading OS from another location. Therefore, ordinary skill would select the boot device not only for loading boot image, but also for loading the OS from the selected device.

4. Claims 4-6, 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee et al (US Patent 6754818), in view of applicant's Admission of Prior Art (AAPA), further in view of Wu et al (US patent 6105130).

For claim 4, Lee et al teach that the boot device is selected when BISST returns a valid device and updates SBDID. However, Lee et al do not teach that the device is selected when "not" bit is set.

Wu et al teach a system where booting is done from a designated device when "yes" is set and from an initially set device when "no" is set (lines 17-27 of column 2 mention that system boots from SCSI device if user input exists, otherwise booting is performed from IDE device. That is equivalent to "yes"/"no" setting)

It would have been obvious for one ordinary skill in the art at the time the invention was made to combine the teachings of Lee et al and Wu et al. One ordinary skill in the art would have been motivated to boot when "no" is set, since that confirms the booting of the system.

For claim 5, the index data in SBDID in Lee is updated if BISST returns valid boot device.

For claims 6 and 16, system is booted in Lee when a good boot device is found.

Response to Arguments

Applicant's arguments with respect to claims 1, 3-18 have been considered but are moot in view of the new ground(s) of rejection. As Lee et al is still relied upon for rejection, Examiner is addressing the relevant logic regarding Lee et al.

Applicant argues that claim 1 recites a system that is multiplexed with a plurality of boot devices in which an OS is stored in each of the boot devices respectively.

Examiner disagrees. Claim 1 does not require each of the plurality of devices to have an OS. Claim 1 requires reading OS from a selected device. Claim1 does not require plurality of OS present in the system.

Applicant argues that the boot devices of Lee of the computer system to be booted do not have an operating system. The computer system is not started up using the operating systems installed in the boot device selected from multiplexed boot devices.

Examiner disagrees. Lines 10-14 of column 3 mention that boot devices 102 contain one or more copies of operating system. 260 of Fig 2 shows the loading of OS. Therefore, it is one of the boot devices 102 that stores the OS that is loaded in step 260. Since the system of Lee permits rotation of boot devices, the device containing OS can be selected as the boot device (as shown in 250 of Fig 2) to load the boot image from and read out the OS from that device. In AAPA, computer system is started up using the operating system installed in the boot device selected from multiplexed boot devices. Thus, Lee et al, in view of AAPA, clearly starts the system using OS in the boot device selected from the multiplexed devices.

Applicant further argues that OSs for starting up the system are not multiplexed in Lee.

Examiner disagrees. There is no requirement in claim that OSs be multiplexed.

Applicant further argues that the boot device to be booted next time is selected from the multiplexed devices regardless whether an abnormality is detected or not in the accessed device.

Examiner agrees that such statement is true for one embodiment. However, claim does not require the selection of boot device from the multiplexed devices only when an abnormality is detected in the accessed device. Therefore, the argument is irrelevant.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Fahmida Rahman whose telephone number is 571-272-8159. The examiner can normally be reached on Monday through Friday 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rehana Perveen can be reached on 571-272-3676. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Fahmida Rahman
Examiner
Art Unit 2116

THUAN N. DU
PRIMARY EXAMINER